NESTAR SYSTEMS, INCORPORATED
CLUSTER/ONE MODEL A (TM)
TECHNICAL NOTE 1

FIXDIR Version 1.0

INTRODUCTION

Intrinsic units may fail in unpredictable ways on the Cluster/One Model A Version 1.0 because they expect unused bytes in a Pascal directory to be zeroed. FIXDIR is a program which zeroes those unused bytes so that intrinsic units work correctly. This should be done for all Pascal volumes created using Version 1.0 of the Cluster/One Model A. This will be done automatically under Version 1.1.

FIXDIR can be executed at any time after a Pascal volume has been created and its directory "zeroed" by the Filer, regardless of the number of files on the volume. The normal sequence is to create and mount a Pascal volume, zero its directory using the filer, execute FIXDIR, and then store Pascal files on the volume.

OPERATING INSTRUCTIONS

1. Execute Pascal:CMD to CREATE a Pascal volume and MOUNT it with read/write access on drive 9, 10, 11, or 12.

   e.g.
   ```
   CREATE pathname,T=P
   MOUNT pathname,D9,RW,EXC
   ```

2. Enter filer and type "Z" to zero directory. In response to the prompt "# of blocks on the disk?," enter the number of blocks used in the corresponding CREATE command. (The default is 280 blocks.)

3. Execute `PASCAL:FIXDIR`

   FIXDIR displays the message: "This program zeroes out the currently unused bytes in the specified directory so that intrinsic units and booting work correctly." It then asks:

   ```
   Unit number to fix (4/5/9...12)?
   ```

   Following the above example, where the disk was mounted on drive (unit) 9, respond: 9. When FIXDIR is done, it returns control to the Pascal command level.

PROGRAM LISTING

```
             (*$C  Copyright 1980, Nestar Systems, Incorporated *)

(* Function:                                                     *)
(*   zeroes the unused bytes in a pascal directory so that      *)
(*   intrinsic units work correctly.  This must be done for all *)
(*   Pascal volumes created using Version 1.0.  This will be    *)
(*   done automatically for volumes created under Version 1.1.  *)

program fixdir;

  var
    i, numfiles, unitnum:  integer;
    disknum:  set of 4..12;
    buffer:  packed array [0..2047] of char;

  begin (*fixdir*)
    writeln ('This program zeroes the currently unused bytes in');
    writeln ('the specified directory so that intrinsic units');
    writeln ('work correctly.');
    writeln;
    disknum := [4,5,9,10,11,12];  (* legal unit numbers for disks *)
    (* get unit number to fix from user *)
    repeat
      write ('Unit number to fix (4/5/9..12) ? ');  readln (unitnum);
    until unitnum in disknum;
    (* get directory into memory                *)
    (* it is 4 blocks long and starts at block 2*)
    unitread (unitnum,buffer,2048,2);
    (* get number of files in this dir *)
    numfiles := ord (buffer [16]);
    (* zero the rest of the entries *)
    for i := (numfiles+1)*26 to 2047 do
      buffer [i] := chr (0);
    (* put the directory back where we found it *)
    unitwrite (unitnum,buffer,2048,2);
  end. (*fixdir*)
```

NESTAR SYSTEMS, INCORPORATED
CLUSTER/ONE MODEL A (TM)
TECHNICAL NOTE 2 (Rev. 0)

Loading DOS 3.3 Disks Into The System

INTRODUCTION

This technical note describes how to create and initialize a virtual disk with a copy of Apple DOS 3.3 from a real minidisk. The created virtual disk can then be used to create other copies and to boot DOS 3.3 directly over the network without using real minidisks. This procedure need only be done once.

This is necesary only for Release 1.0 of the Nestar File Server, which was distributed before the availability of DOS 3.3. Subsequent releases of the File Server will have DOS 3.3 already loaded.

COMPATIBILITY BETWEEN DOS 3.2 AND DOS 3.3

Unlike real Apple minidisks, initialized virtual DOS disks of either type can be used with either version of DOS. This means that there is no requirement to convert existing virtual DOS 3.2 disks to DOS 3.3 format, except to take advantage of the larger size. For real disks, each version of DOS only supports its own format.

Care must be taken when INITing a virtual disk, so that you use DOS 3.2 to INIT virtual SECTORS=13 disks, and DOS 3.3 to INIT virtual SECTORS=16 disks. A mistake will either cause some sectors to be inaccessible (DOS 3.2 with a SECTORS=16 disk), or result in drive errors as DOS attempts to access non-existent sectors (DOS 3.3 with a SECTORS=13 disk).

OVERVIEW

The basic sequence is

1. Boot from a local real minidisk which contains a copy of DOS 3.3,
2. Patch the copy of DOS in memory so that it can access the network for commands and virtual disk I/O, then
3. Copy the real minidisk to a virtual disk.

Detailed instructions for doing each of these steps are given in the following section. Experienced users, however, may find that this overview is sufficient, given the fact that the appropriate patch for step 2 is BD00:20 1F C5 EA for a 48K Apple with the network card in slot 5.

DETAILED INSTRUCTIONS

1. THE HARDWARE CONFIGURATION. With the Apple user station powered off, install the Nestar network interface card in slot 5, and a 16-sector Apple minidisk controller in slot 6, with an Apple disk attached as drive #1.

2. BOOT FROM THE LOCAL DISK. Insert a DOS 3.3 master disk in the Apple disk drive. Turn on the Apple to cause DOS to boot from the minidisk. If you do not have an autoboot ROM, you will need to type

```
6<CONTROL-P><RETURN>
```

3. PATCH DOS TO ALLOW NETWORK COMMANDS. Get to the monitor by typing

```
CALL -151 <RETURN>
```

Verify the data which will be patched by typing

```
BD00.BD03 <RETURN>
```

and check that the correct data are displayed as follows:

```
BD00- 84 48 85 49
```

Carefully patch in the new data by typing

```
BD00:20 1F C5 EA <RETURN>
```

The above addresses assume that the Apple has 48K bytes of memory (not counting memory on the language card, if installed). If your Apple has only 32K, then the above addresses in the form "BDxx" should be "7Dxx" instead. If your Apple has only 16K, then those addresses should be "3Dxx". In all cases, if you cannot verify the old data, then you should not proceed with the rest of the instructions. Recheck the memory size of your Apple, and verify that you are booting a DOS 3.3 master disk. If you still cannot verify the proper data, you should get further assistance.

4. PREPARE THE VIRTUAL DISK (CLUSTER-FILE). Open a channel to issue network commands to the file server by typing

```
PR#5<RETURN>
```

Create an appropriate virtual disk using the create command (see the Cluster/One Model A User Manual for more details). A typical command might be

```
@CREATE /MAIN/DOS/3.3,T=D,SECTORS=16 <RETURN>
```

Now mount both the newly created virtual disk, and an existing DOS virtual disk which contains a copy of NESTAR DISK COPY. Commands to do that might be

```
@MOUNT /MAIN/DOS/3.3,D1,RW,EXC <RETURN>
@MOUNT /MAIN/DOS/3.2.1,D2 <RETURN>
```

(The pathnames of these virtual disks may be different at your installation.)

5. COPY FROM THE REAL MINIDISK TO THE VIRTUAL DISK. Run the Nestar Disk Copy program by typing

```
BRUN NESTAR DISK COPY,D2,S5 <RETURN>
```

When it asks DOS OR PASCAL, specify D for DOS.
When it asks what the ORIGINAL disk is, specify slot 6, drive 1.
When it asks what the COPY disk is, specify slot 5, drive 1.
When it asks whether to INIT the copy disk, specify Y.
When it asks for the volume number, specify 254 <RETURN>.

The entire original minidisk will then be copied to the virtual disk. In addition to copying DOS 3.3, all user files will be copied.

To verify that the copy is successful, turn off the Apple, remove the minidisk controller, and turn the Apple back on. Respond to the "Volume to boot?" question with the name of the newly created DOS 3.3 virtual disk. DOS 3.3 will be booted from the File Server, and the HELLO program, if any, will be run.

NESTAR SYSTEMS, INCORPORATED
CLUSTER/ONE MODEL A (TM)
TECHNICAL NOTE 3 (Rev. 0)

ATTACHING A SORRENTO VALLEY DISK SYSTEM TO A CLUSTER/ONE MODEL A USER-STATION

INTRODUCTION

The standard Cluster/One Model A system allows a user to read and write on an Apple mini-floppy disk unit attached to a standard Apple disk controller card inserted in one of the expansion slots of the Apple. The following changes allow a user to substitute a Sorrento Valley 8-inch floppy disk unit and corresponding disk controller card for the mini disk and standard disk controller card.

The program assumes:

1. The disk controller card is in slot 7.
2. The network interface card is in slot 6.

Depending on the intended use, the user may add these changes to DOS in a number of ways:

1. Booting a DOS volume which already contains the desired DOS changes.
2. Executing a program that causes the desired changes in DOS.
3. Manually changing DOS each time, using the Apple monitor.

REQUIRED DOS CHANGES

1. Boot the Sorrento Valley disk system in slot 7 using their standard procedures.

2. Enter the Apple monitor:

```
]CALL -151
```

3. Verify the area that is to be changed:

```
*BD00L
```

NOTE: The above address is correct for a 48K Apple. If your Apple has 32K of RAM, you should get further assistance from your dealer or Nestar Systems.

You should see one of the following:

```
BD00-   2C FF CF        BIT $CFFF
BD03-   2C 00 C7        BIT $C700
BD06-   4C D2 C9        JMP $C9D2
          etc.
```

or

```
BD00-   2C FF CF        BIT $CFFF
BD03-   2C 00 C7        BIT $C700
BD06-   4C E0 C9        JMP $C9E0
          etc.
```

or

```
BD00-   2C FF CF        BIT $CFFF
BD03-   2C 00 C7        BIT $C700
BD06-   4C 18 C8        JMP $C818
          etc.
```

IF YOU DO NOT SEE ONE OF THOSE, DO NOT PROCEED. Contact your dealer or Nestar Systems for further assistance.

4. You are going to change two areas: BD00, and BC00.

Enter:

```
*BD00:4C 00 BC

*BC00:20 1F C6 A4 48 A5 49 2C FF CF 4C 03 BD
```

Verify these changes, by:

```
*BD00L
```

and seeing:

```
BD00-   4C 00 BC        JMP $BC00
          etc.
```

and

```
*BC00L
```

and seeing:

```
BC00-   20 1F C6        JSR $C61F
BC03-   A4 48           LDY $48
BC05-   A5 49           LDA $49
BC07-   2C FF CF        BIT $CFFF
BC0A-   4C 03 BD        JMP $BD03
          etc.
```

If these are correct, enter <ctrl-B> to reenter BASIC. You can now enter

```
]PR#6
]@MOUNT something,D1
```

and do a CATALOG S6,D1 to see the CATALOG of the virtual disk mounted by the previous command, and CATALOG S7,D1 to see the CATALOG of the disk mount in drive 1 of the Sorrento Valley disk system.

NOTE: The above sequences of changes adds code at location BC00. This code will be destroyed if you attempt to read or write to a standard Apple minidisk. This set of changes will permit you to use both the Sorrento Valley disks and the Nestar File Server, but not the Apple minidisks concurrently.

5. If you wish, you can convert the above set of changes into decimal, and use POKEs from BASIC to change the two sets of locations from a program. The HELLO program is one place to put this, so that it will happen everytime you boot the Sorrento Valley disk. Alternatively, you can put it in a program that you run by a command, such as:

```
]RUN NESTAR LINK
```

whenever you want to perform the linking.

NESTAR SYSTEMS, INCORPORATED
CLUSTER/ONE MODEL A (TM)
TECHNICAL NOTE 4 (Rev. 0)

Summary of space requirements for NFS tables
Release 1.0 and Release 1.1

INTRODUCTION

This note attempts to summarize the space required in the File Server Apple for various tables, etc. It is only an approximation, and may be refined in the future.

SPACE IN RELEASE 1.0

There are five types of dynamically created objects. Each one has its own space requirements. A free-list chain is kept for each of these five types. When an entry is needed, one is taken from the free-list, if available. When an entry is freed up, it is returned to the free-list, and becomes available for future requirements for that specific type of object. When an entry is needed, and the corresponding free-list is empty, new space is allocated from the heap. Prior to doing this, a check is made to ensure a certain amount of "breathing room" will be left after allocation. If there is not sufficient room, a "memory full" error message is generated.

LOCK: Approx. 24 bytes are required for each active lock. These objects are not shared among users (even for SHRed locks).

VCON: Approx. 10 bytes are required for every active MOUNT. (Note: additional space is required for VDCB and FCBs below).

LOG: Approx. 12 bytes are required for every station that has a non-null DIR, PRVPW or GRPPW set.

VDCB: Approx. 16 bytes are required for every active MOUNT. Users which have the same disk MOUNTed share the same VDCB.

FCB: Approx. 142 bytes are required for every active MOUNT or SET DIR. Users which have the same disk MOUNTed share the same FCB. Each SET DIR requires its own FCB.

Notes: The size of control blocks are fixed, and set for the maximum length of locknames, pathnames, etc. Using shorter names does not affect storage requirements.

The most expensive (in terms of space) objects are FCBs. Clearly you should not have MOUNTs (or SET DIRs) to disks which aren't needed. MOUNT statements should be issued dynamically as required.

TOTAL SPACE available for tables: Approx. 7200 bytes

This provides room for about 43 unshared mounts (7200 / (10+16+142=168).

SPACE IN RELEASE 1.1

There are several changes in Release 1.1 which attempt to avoid certain space anomalies. In this Release, space for LOCKs, VDCBs, VCONs and LOGs are all allocated in units of approx. 18 bytes. VDCBs, VCONs and LOGs require one 18 byte unit, whereas each LOCK requires two of these units. The space management algorithms can reuse the space freed up by any of the four types of objects. Thus using up all the space by doing LOCKs and then freeing them will permit that space to be used for VCONs, etc.

In the case of SET DIRs to the same cluster file, the FCBs are shared under Release 1.1. Note, however, that separate FCBs are required for MOUNTs and SET DIRs even if to the same pathname. (This is very unlikely to be relevant, since SET DIRs are always to directories, and MOUNTs are generally to virtual disks.)

LOCK: Approx. 36 bytes are required for each active lock. These objects are not shared among users (even for SHRed locks).

VCON: Aprox. 18 bytes are required for every active MOUNT. (Note: additional space is required for VDCB and FCBs below).

LOG: Approx. 18 bytes are required for every station that has a non-null DIR, PRVPW or GRPPW set.

VDCB: Approx. 18 bytes are required for every active MOUNT. Users which have the same disk MOUNTed share the same VDCB.

FCB: Approx. 28 bytes are required for every active MOUNT or SET DIR. The sharing rules are the same as for VDCBs.

Notes: Additional space is required for a pathname if Release 1.1 cannot find the file /MAIN/SYSTEM/PATHNAMES. If not present, each FCB takes as additional approx. 82 bytes.

TOTAL SPACE available for tables: Approx. 5600 bytes

For example, if the SYSTEM/PATHNAMES file exists, this provides room for about 88 unshared mounts (5600 / (18+18+28=64). If the SYSTEM/PATHNAMES file does not exist, this provides room for about 38 unshared mounts (5600 / (18+18+28+82=146).

NESTAR SYSTEMS, INCORPORATED
CLUSTER/ONE MODEL A (TM)
TECHNICAL NOTE 5 (Rev. 0)

NETLINK Version 1.0

INTRODUCTION

This program is used to permit a user station which has booted Apple Pascal locally from Apple minidisks to access the Cluster/One Model A network.

Apple Pascal, as distributed by Apple Computer, Inc., can access data stored on (real) minidisks. To permit it to access data stored on the Nestar File Server disks, certain modifications are made to the Pascal system as it is read into a user station by the BOOT program (when booting Pascal from the Nestar File Server). The NETLINK program makes these same modifications, but does so after Apple Pascal has been booted from a (local) minidisk.

Users who choose to regularly use their user stations in local mode, and only rarely utilize the network can use this program. Since the Apple Pascal system requires that you boot from slot 6, use of NETLINK avoids the need to move controller cards around. Simply leave the Apple in its normal configuration, and put the Nestar network interface card in slot 1, 2 or 3.

OPERATING INSTRUCTIONS

1. To execute NETLINK, and access the network, you must first have a standard Apple minidisk with two programs on it:

   NETLINK.CODE and
   CMD.CODE

   These programs are distributed to your installation as part of the disk volume /MAIN/PASCAL/UTILS, and should have been copied to the user accessible volume /MAIN/PASCAL/LIB by your System Manager. He can tell you how to access these programs, and use the Filer to copy them onto a real minidisk.

2. You should now boot your Apple into Pascal, using a minidisk controller in slot 6. The Nestar network interface card should be placed into slot 1, 2 or 3. The Pascal system will NOT boot if the Nestar network interface card is placed in slot 4 or 5.

3. Now X(ecute the program NETLINK from the minidisk. The program will display the message:

   ```
   Pascal now linked to Cluster/One network
   ```

   and exit to the command level of Pascal.

NETLINK makes a test to see that a compatible version of the Pascal system is running before making any changes. If it does not support or recognize the version that is running, it will display the message:

```
This version and flavor of Apple Pascal
    not supported.
No linking has been performed.
```

and exit without making any changes.

NOTE: At the time of publication of this Technical Note, the only version of Apple Pascal supported by NETLINK is Apple Pascal Release 1.1.

4. You should now X(ecute the CMD program, and use it to set up any virtual drives mounts, etc. you wish. Do not change drive 4 from real to virtual under any circumstances!

   For example, when you issue the command

   ```
   SHOW DRIVES
   ```

   you will see that initially all of the disk drives are considered REAL.

   You must first issue the command

   ```
   SET DRIVE 11,12 VIRTUAL
   ```

   for example. Next you can MOUNT virtual disks on D11 and D12, and access them from Pascal, such as:

   ```
   MOUNT /MAIN/PASCAL/LIB,D11
   ```

   Note that drives 9,10,11 and 12 can be made virtual even if a real Apple minidisk controller is installed. The system will then access the corresponding virtual drives rather that the physical ones in slots 4 and 5. You can switch back as well (e.g. by executing CMD and entering SET DRIVE 11 REAL). As mentioned earlier, you must NOT set drive 4 to virtual, due to the internal nature of the Pascal system.

NESTAR SYSTEMS, INCORPORATED
CLUSTER/ONE MODEL A (TM)
TECHNICAL NOTE 6 (Rev. 0)

ClusterBus Cabling Information

INTRODUCTION

This note is a description of the electrical requirements for the communications bus which connects together a collection of Cluster/One Model A stations. It provides information on a variety of useful connectors which can be used to construct a network tailored to an installation's special requirements. A variety of standard cables are available already assembled from Nestar; contact the Sales Department for information.

GENERAL INFORMATION

The Nestar Model A bus interface card (P/N 1015-2) provides the electrical and logical interface between the Apple II computer (via its 8 peripheral slots) and the interstation communication bus. Looking at the interface card, you will see two sets of 16 pin headers. Each header has two rows of eight pins, with 0.1" spacing. These two headers are identical; all sixteen pins are connected together in parallel.

Two headers are provided in order to simplify a very common case in many standard Model A networks, namely the ability to daisy chain several nearby stations together. This is accomplished by bringing the bus to one card, and then running a length of cable from that card to another using the second header, and so on down the line. Other configurations can be accomplished using other collections of connectors and splices which are described later in this Note.

In short, a Nestar Model A network simply requires that all stations must connect, in parallel, to a common 16 wire bus.

The bus does not constrain the manner in which you connect all stations in parallel. They may be daisy chained, but this is not required. Think of the AC power distribution in your house or office; just as the ground and hot wires must ultimately go to every light and outlet using any combination of tees and chains, so must the 16 wires go to every Nestar bus interface.

WARNING

The speed, delays and timing of the Nestar interface has been carefully designed to permit a very broad set of possible connection configurations. This design allows for reflections caused by unterminated lines, mismatches at tees and splices etc. Nestar has tested the system extensively at a large variety of "worst case" configurations, using at least one thousand feet of standard 16-wire ribbon cable. It is not possible to test every conceivable configuration (since the number is unbounded!). Should you encounter some situation, which does not function within the published restriction to one thousand feet, contact Nestar Systems.

ELECTRICAL PROPERTIES

The Nestar Model A bus is driven by open collector high current TTL drivers. The voltage levels on the bus switch between 0 and +5 volts. The normal resting level for signal wires is +5 volts, and is lowered to 0 by the driving station. The drivers were chosen so that a station which is powered off does not affect the bus. Thus only those stations which are being used need be turned on.

In order for such a bus to function, and rest at +5 volts, one station on the network must "pullup" the bus; this is normally done by the effect of a resistor network installed (in socket RN2) on the bus interface card of the Nestar File Server. Because of this, the File Server must be powered on for the bus to function.

Actually, communications can take place between ANY two stations on the network. The File Server need not be part of this communications; however it must be powered on to provide the "pullup" function described above.

BUS SIGNAL NAMES AND POSITIONS

16 pin header	ribbon wire no.	signal name	DB-15 pin no.
A	(stripe) 1	Data 4	(stripe) 1
1	2	Data 0	9
B	3	Data 5	2
2	4	Data 1	10
C	5	Data 6	3
3	6	Data 2	11
D	7	Data 7	4
4	8	Data 3	12
E	9	GND	5
5	10	HS1	13
F	11	GND	6
6	12	HS2	14
H	13	GND	7
7	14	HS3	15
J	15	GND	8
8	16	UNUSED	

The left hand side of the above diagram shows the pin identifications on the sixteen pin header on the Nestar interface, and the connection made to a standard 0.05" spaced ribbon cable.

This wire/signal relationship will be maintained throughout any configuration built from standard IDC (insulation displacement connector) family of connectors, tees, etc.

The right hand side shows the signal connections made when going from 16 wire ribbon cable to an industry standard DB-15 connector. This adaption can be done most easily using an IDC component, such as AMP P/N 207331-1 or equivalent.

NOTE: Be sure to use a connector which omits wire 16 when going from 16 wire ribbon to 15-conductor cable.

The pin assignments have been chosen so that there are ground wires surrounding all the handshake (HS) signals when a ribbon cable is used. When using a cable with discrete wire twisted pairs, each of the handshake signals should be twisted with a ground wire. For example, the cable which terminates onto the male DB-15 connector should have 7 twisted pairs, with pairs going to pins 1-9, 2-10, 3-11, 4-12, 5-13, 6-14, and 7-15. A separate drain wire, and the outer shield should be connected to pin 8, a ground.

USEFUL IDC PARTS IN BUILDING A BUS

The following connectors are of interest in configuring a Nestar communications bus.

name	sample supplier P/N	typical usage
16 pin female socket	AMP 88011-1	connection to card, from flat cable
16 pin male tap	AMP 88357-5	use on main bus, to "tee" drops to stations
DB-15 female adaptor	AMP 207331-1	interface flat cable to DB-15 connectors
DB-15 male connector	Cannon DB-15P	round cable to DB-15 connector

The following cables are of interest in configuring a Nestar communications bus.

MODEL A CABLES

name	sample supplier P/N	usage
16 wire ribbon (#28)	3M Scotchflex 3365/16	standard applications
jacketed 16 wire ribbon	3M Scotchflex 3603/16	better for pulling through trays, or to prevent damage from stepping on, bending, etc.
16 wire ribbon with ground plane	3M Scotchflex 3469/16	better signal propagation and noise immunity
twisted pairs with drain and shield	Belden 9508	round cable, low noise

TOOLS FOR CABLING

There is a great temptation to try and assemble IDC components using handy things like vises, pliers and hammers. Unless you are a mechanical Wizard, DO NOT DO THIS!! The chances of ruining the connectors, not making a good connection, or shorting the cable are EXTREMELY high.

You should either use the proper presses and tools, such as the 3M Scotchflex assembly press (Model 3640 or Model 3335), or else have cables and adaptors made by a firm which has the proper assembly or molding equipment. Much time and energy can be spent after using improper equipment trying to track down a problem it caused.

BUILDING A MAIN BUS

In certain configurations, the simplest way to wire the system is to start from the File Server, and daisy chain standard lengths of 16 wire ribbon assembly from station to station. Alternatively, the main bus can be pre-wired and placed in a ceiling or cable runs above the user area. Separate drops can then be run from the ceiling area, down inside walls if desired, exiting near each user station. Nearby user stations can be daisy chained together.

We recommend constructing a main bus as follows: build separate segments from 16 wire ribbon cable, each 50 to 100 feet long. At one end, use a 16 pin female header. At the other, use a 16 pin male tap. In between, at appropriate places apply a 16 pin male tap.

OTHER USEFUL STANDARD ASSEMBLIES

There are two handy cable adaptors which you might consider using in constructing or modifying a Nestar network.

The first of these is a short length of 16-wire ribbon cable, which has a 16-pin female connector at one end, and two 16-pin male taps near the other end. This configuration is useful for adding an additional spur or drop to an existing main bus scheme (as described in the previous section). Rather than having to apply an additional male tap to a bus which is installed and operational, it is suggested that this adaptor be built and tested independently. It can then be plugged into a male tap on an existing bus, thus providing the original tap position, plus one more, without potentially damaging the bus installation. Such an adaptor may also be useful when attached to user interface cards in a daisy-chained environment, in order to provide bus continuity when removing an interface card for repair, etc.

A similar adaptor, but using two DB-15 connectors in place of the 16-pin male taps is appropriate at each user station when interconnecting stations with cables terminated in DB-15 connectors rather than female ribbon connectors. Again, placing two connectors on one short length will provide for bus continuity even if an interface card is removed from the network.

NESTAR SYSTEMS, INCORPORATED
CLUSTER/ONE MODEL A (TM)
COPYRIGHT 1981
TECHNICAL NOTE 7 (Rev. 0)

NETUNIT UNIT for NFS Version 1.1 and 1.1.1

INTRODUCTION

The NETUNIT unit allows the Pascal user to write programs which issue commands to a Network File Server (NFS). Lower Level communication routines are also included to allow the program to send network messages to any station and to receive network messages from other stations.

The NETUNIT unit is a Pascal "regular unit" and requires the external assembly language routines NETRTNS (NWSELECT, etc.) and NETASM. These are all contained in the library:

PASCAL:NETWORK.LIBRARY

ICS

In order to communicate with another station on a Nestar Network, two pieces of information are required to identify the station, namely:

slot - The slot number of the network card in your Apple which identifies the network to be used. (Remember, your Apple can be connected to multiple Nestar Networks by having multiple network cards in different slots.)

station - The station number of the station you wish to communicate with.

This information is called the "virtual channel" or just "channel". It describes the path used to get a message from your station to another station.

When using the network interface routines in NETUNIT the channel information must be supplied. There is a default channel which is used if a channel is not explicitly specified. This is actually the default NFS channel. Initially, the default NFS channel is set to the slot and station number of the NFS you booted from. (The "SET NFS" local command can be used to change this value.)

For the "NFS" routines in NETUNIT, where a pointer to the channel is expected, "NIL" can be specified to indicate the default NFS channel is to be used.

For the "BUS" routines in NETUNIT, where the channel itself is specified, the slot number in the channel passed to the routine; can be set to zero and the default NFS channel slot number will be used.

GENERAL ORGANIZATION

The NETUNIT unit consists of the following Pascal procedures/functions:

1. Network File Server (NFS) Communication Routines

NFSCMD - Send command to NFS and write the responses to the screen. (The routine will retry if the network is not immediately available or if there is a network error.)

NFSCMDO - Send command to NFS and write the responses to the screen. (The routine will NOT retry if the network is not immediately available or if there is a network error.)

NFSCMD1 - Sends command to NFS. Successive calls to this routine return each line of the response to the program. (The routine will NOT retry if the network is not immediately available or if there is a network error.)

2. Low Level Communication Routines

BUSSEND - Routine to send an arbitrary message to another station on the network.

BUSRCV - Routine to receive a message from the network.

3. Local Information Routines

CHANINFO - Routine to return the channel information for a Pascal unit. (See "BASICS" section for a description of channels.)

CHANSET - Routine to set the channel information for a Pascal unit. (See "BASICS" section for a description of channels.)

UNITINFO - Routine to return general information about the status of a Pascal unit (e.g., volume name). (See the UNITDESC record in the TYPE section of the NETUNIT INTERFACE listing which is provided for the format of the information.)

4. Global (INTERFACE) variables

The global variables maintained by NETUNIT (defined in the INTERFACE section of the unit), which may be useful to programs using this unit are:

PVERS	(integer)	- Pascal version number (0 = version 1.0, 2 = version 1.2 -- as defined by Apple).
PFLAV	(integer)	- Pascal flavor (0 = regular -- as defined by Apple).
NETPRES [0..7]	(boolean)	- Shows the presence or absence of a network card in each slot.
MYSTN [0..7]	(integer)	- Our station number for the network card in each slot, or zero if it is not a network card.
NETERR	(integer)	- Low level network return code (indicates the reason for a network error, e.g. collision, bus busy, etc.).

OPERATING INSTRUCTIONS

The following is the INTERFACE section of the NETUNIT unit. It describes the format of the parameters for all the routines.

```
INTERFACE

TYPE
     CHANNELTYP = PACKED RECORD
                    SLOT : 0..255;
                    STN  : 0..255;
                    PORT : INTEGER
                  END;

     CHANNELPTR = ^CHANNELTYP;

     UNITDESC = PACKED RECORD
                  UNITVID: STRING [7];  (* volume id *)
                  UNITBLKF: BOOLEAN;  (* unit is "block structured"--a disk *)
                  UNITLSTB: INTEGER;  (* highest block number on device --
                                         for block structured units only--
                                         $7fff if not yet known *)
                  UNITVIRF: BOOLEAN;  (* unit is virtual *)
                  UNITCHAN: CHANNELTYP  (* virtual channel description *)

     NETLONGSTR = STRING [255];

VAR
  PVERS,PFLAV : INTEGER;          (* Pascal version and flavor *)

  NETPRES  : ARRAY [0..7] OF BOOLEAN; (* shows presence or absence of net card
                                         ... in each slot *)
  MYSTN    : ARRAY [0..7] OF INTEGER;  (* station numbers for nets *)

  NETERR   : INTEGER;  (* low level "nwret" error code *)
```

```
PROCEDURE NWSELECT(SLOT :INTEGER);          (*  Low       *)
PROCEDURE NWINIT;                           (*  Level     *)
PROCEDURE NWCHK ;                           (*  Nestar    *)
PROCEDURE NWRCV ;                           (*  Network   *)
PROCEDURE NWSEND;                           (*  I/O       *)
PROCEDURE NWCONN;                           (*  Routines  *)
PROCEDURE NWDISC;
PROCEDURE NWGMSG;


FUNCTION NFSCMD   (    CHANPTR: CHANNELPTR;
                           CMD: NETLONGSTR;
                       VAR RSP: STRING): INTEGER;


FUNCTION NFSCMDO  (    CHANPTR: CHANNELPTR;
                           CMD: NETLONGSTR;
                       VAR RSP: STRING): INTEGER;


(* NFSCMD and NFSCMDO send NFS commands to the NFS designated by   *)
(* channelptr.  If a nil channelptr is specified, then the         *)
(* default NFS is assumed.                                         *)
(*                                                                 *)
(* NFSCMD retries if there is a network error (such as             *)
   "bus busy"), and NFSCMDO does not.                              *)
(*                                                                 *)
(* Returns final line of response in RSP and nfs error             *)
(* code in function result  (eg RSP = 'OK' and NFSCMD = 0          *)
(* having printed intermediate response lines to screen            *)
(* (to intercept intermediate lines use NFSCMD1).                  *)
(*                                                                 *)
(* Code to use NFSCMD looks like:

          USES NETUNIT;
          ...
          CMD := 'command';
          rcode := NFSCMD(nil,cmd,rsp);
          process RSP;
          ........................................*)
```

```
FUNCTION NFSCMD1 (CHANPTR  : CHANNELPTR;
                      CMD  : NETLONGSTR;
                  VAR RSP  : STRING;
                   NEWCMD  : BOOLEAN;
                 VAR DONE  : BOOLEAN): INTEGER;

(* Use NFSCMD1 to intercept screen displayed responses.               *)
(* (Use NFSCMD for sending NFS commands unless it is necessary        *)
(* to intercept NFS responses that are normally displayed             *)
(* directly on the local screen without program intervention.)        *)
(*                                                                    *)
(* NFSCMD1 sends NFS commands to the NFS designated by                *)
(* channelptr.  Called with newcmd = true for initially sending       *)
(* cmd to NFS, then iterate calling NFSCMD1 with newcmd=false         *)
(* until done=true.  When done = true rsp contains the last text      *)
(* rec'd from network (e.g., OK), or local network error msg          *)
(* and NFSCMD1 = return code (e.g., 0).                               *)


(* code to use nfscmd1 looks like:

            USES NETUNIT;
            ...
            CMD := 'command';
            rcode  := NFSCMD1(nil,cmd,rsp,true,done);
            WHILE NOT DONE DO
            BEGIN
              process RSP; (rcode not set until DONE=true)
              rcode  := NFSCMD1(nil,cmd,rsp,false,done);
            END;
            process final RSP from nfs if desired;
            ..............................................*)
```

```
FUNCTION BUSSEND
          (CHANNEL  : CHANNELTYP;
           MSGADDR  : INTEGER;   (*..msg : string;..*)
           MSGTYPE,LEN:INTEGER):INTEGER;

(* Sends string MSG of length LEN bytes                            *)
(* of type MSGTYPE (message type is interpreted by the             *)
(* sender and receiver by mutual convention, however               *)
(* message types 0 to 127 are reserved by Nestar, their            *)
(* use may cause unpredicable results)                             *)
(* to channel CHANNEL.  function value is network return code *)
(* (bussend = 0 if message transmitted ok)                         *)


FUNCTION BUSRCV
          (VAR CHANNEL  : CHANNELTYP;
                MSGADDR : INTEGER;
           VAR MSGTYPE  : INTEGER;
                    LEN : INTEGER;
               VAR TLEN : INTEGER): INTEGER;

(* Receives message into string MSG                                 *)
(* of maximum length LEN bytes (TLEN = actual length rec'd)         *)
(* of type MSGTYPE from CHANNEL.  Function value is network         *)
(* return code (busrcv = 0 if message transmitted ok)               *)
```

```
PROCEDURE CHANINFO (UNITNO: INTEGER; VAR CHAN: CHANNELTYP);

(* CHANINFO returns information about the virtual channel connection
   for the specified Pascal UNITNO (1..12).  If a UNITNO of 0 is given
   then the NFS channel connection information is returned.

   If a channel of zero is returned then the device is REAL (i.e.,
   has no virtual channel associated with it).  *)


FUNCTION CHANSET (UNITNO: INTEGER; CHAN: CHANNELTYP): INTEGER;

(* CHANSET sets the specified Pascal UNITNO (1..12) or NFS (if UNITNO=0)
   virtual channel connection to the channel specified.  Note that this
   will have the effect of setting a REAL device to virtual.

   If a channel of zeros is given, then the device will be set to REAL
   (i.e., no virtual channel connection).

   CHANSET returns a return code of...
       0         - channel set
       pos       - error code (see network error codes) *)


 PROCEDURE UNITINFO (UNITNO: INTEGER; VAR U: UNITDESC);

 (* UNITINFO returns all the information available about a Pascal
    unit (1..12).  This includes the volume name, and the virtual
    channel path (or zeros if real). *)
```

EXAMPLE PROGRAM

The following program, MINICMD, demonstrates the use of NFSCMD to send commands to a NFS with a Pascal program. The MINICMD program will get an input line from the user and send it to the default NFS for execution.

The program listing is as follows:

```
PROGRAM MINICMD;

USES (*$U MODAOBJ:NETWORK.LIBRARY *) NETUNIT;

VAR
  RC: INTEGER;
  CMD,REPLY: STRING;

BEGIN
  WRITELN ('Mini Command Program');
  REPEAT
    WRITE ('NFS? '); READLN (CMD);
    IF LENGTH (CMD) <> 0 THEN BEGIN
      RC := NFSCMD (NIL,CMD,REPLY);   (* Send command to default NFS *)
      WRITELN (RC,',',REPLY)
    END;
  UNTIL LENGTH (CMD) = 0;  (* All done if nothing typed *)
  WRITELN ('Ciao.')
END.
```

A sample execution of the MINICMD program is as follows:

```
Mini Command Program
NFS? show date
DATE IS 03/10/81
NFS? foo
1,ILLEGAL COMMAND
NFS?
Ciao.
```

NOTES

There are other related units which use NETUNIT in NETWORK.LIBRARY, they are:

CMDUNIT	- Provides the support for local commands (e.g., SET DRIVE, SET NFS, etc.) in addition to NFS commands. This is the unit that the CMD utility program uses.
TALKUNIT	- Provies a higer level support for station to station communications than BUSSEND and BUSRCV routines. For example, it provides a message buffering mechanism.

Refer to the documentation of these units for more information.

NESTAR SYSTEMS, INCORPORATED
CLUSTER/ONE MODEL A (TM)
TECHNICAL NOTE 8 (Rev. 0)

Hard Disk Power Sequencing

INTRODUCTION

Users of Hard Disks should follow procedures when powering the Network File Server on or off. Incorrect procdeures can cause loss of data, and may make the File Server incapable of booting.

OPERATING INSTRUCTIONS

1. To power on the system, first power on all hard disks, with the NFS Apple powered off. Then power on the NFS Apple to give the hard disk a reset pulse. After 3 minutes, the hard disks should have finished the power on sequence. Now, power off the Apple, and power on again. The hard disk should boot.

2. To power off the system, first halt NFS using the QUIT command from the console. Then, eXecute HDISK. Specify the PD option to power down the hard disks. First power down the hard disk not booted from. Then power down the boot disk. After the SYSTEM STOPPED message is displayed, turn off power at the hard disks using the switches in back of the units. Then, power down the NFS Apple.

NESTAR SYSTEMS, INCORPORATED
CLUSTER/ONE MODEL A (TM)
TECHNICAL NOTE 9 (Rev. 0)

HARD DISK BOOT PROCEDURE

INTRODUCTION

Some new M40 And M20 hard disk units have engineering changes which change the timing of the reset function. The effect of this change is that the hard disks will no longer autoboot with B1.1 ROMs and Revision C of the hard disk controller. This change is in effect for the serial number range of #1176 and up. This number is on the specifications plate found on the front of the hard disk unit, behind the movable face plate.

OPERATING INSTRUCTIONS

1. To boot a hard disk which has this problem, first power up the hard disk. The disk may require the Apple to be powered on in order to complete the power on and self test sequence. This sequence takes about three minutes, during which time the disk may emit rumbling noises.

2. After three minutes from power on, the Apple should be powered off. Then powered on again, it will go through a sequence looking for UNIT 05, UNIT 06, etc. Do NOT allow it to try to boot from the hard disk the first time, but hit <esc> instead.

3. The second time the program tries to boot from UNIT 05, allow it to boot.

In short, power on the hard disk, power on the Apple, then wait for three minutes. Power the Apple off then on and press <esc> (escape) as soon as you see "UNIT 05" on the screen.

NESTAR SYSTEMS, INCORPORATED
CLUSTER/ONE MODEL A (TM)
TECHNICAL BULLETIN 1 (Rev. 0)

DISK MAINTENANCE Version 1.0

Hard Disk And Floppy Disk Drive Preventive Maintenance

INTRODUCTION

The Hard Disk Drive requires very little preventive maintenance. Corrective maintenance is limited to failure isolation and parts replacement. Periodically, approximately every 60 days, the unit should be checked for the following items:

INSTRUCTIONS

1.0 Hard Disk Preventive Maintenance

(NOTE: TO PERFORM ANY OF THE FOLLOWING THE POWER MUST BE REMOVED. equipment described in this bulletin contains hazardous voltages capable of inflicting personal injury. The cautionary and warning notes are included in this bulletin to alert operator and maintenance personnel to these possible hazards.)

1.1 Clean the air filter. It is located in the front of the disk. Remove the front panel by lifting up and you will have access to the filter. Remove the two screws and the filter will come off. Using a vaccum or air hose remove the dust from the filter. If there is an excess of dirt it is suggested you move the system to an area that is relatively dust free. There is little chance of failure from the dust, but build-up could eventually cause a failure. (See Diagram 1.0.)

1.2 Remove dust or oil from spindle ground strap. The spindle ground strap is located on the bottom of the unit. You will have to turn the unit on its side to perform this particular function. (NOTE: This is to be done with the utmost of caution. Be careful as not to drop the unit. You may cause an uncalculated amount of damage if dropped.) Remove the two visible screws and you will have access to the bottom of the unit. The spindle ground strap is a copper strap resting on the timing wheel. Gently lift the ground strap and using a soft cloth wipe off any dust or oil from the ground spindle contact head. Replace the head on the timing wheel, insure it is making firm contact with the timing wheel. (See Diagram 1.1.)

1.3 Removal of dust from Timing Wheel Disk. The possibility exists that there may be a buildup of dust particles on the surfaces of the Timing Wheel Disk due to static electricity. If this buildup occurs, problems associated with reading and writing particular sectors may be experienced intermittently. This may be corrected by periodically cleaning the Timing Wheel Disk. The Timing Wheel Disk is located just above the spindle pulley on the bottom of the drive. Use a soft bristled brush to clean the disk. (See Diagram 1.2.)

1.4 Measure the +5 Volts on the Power Supply and insure it is within 1%. To measure this voltage you must remove the enclosure. This measurement must be obtained with the power on. Be extremely careful. Only qualified personnel should endeavor to make this measurement. This measurement may be taken on either the Intelligent Marksmen Controller PCB or the Basic Drive Control PCB´s. The voltage points are clearly marked as +5v. You may use any point clearly marked as DC Ground as a ground return point. (See Diagram 1.3.)

INTRODUCTION

The Floppy Disk Drive has been designed to keep maintenance as simple and infrequent as possible. Under normal circumstances preventive maintenance should be performed approximatetly every 60 days. If severely dirty environments are encountered, an occasional cleaning of the drive may be performed to assure continued reliable performance. If a drive malfunctions, it is recommended that it be inspected and cleaned as described below. Section 2.1 lists the maintenance equipment required for the various procedures. The remaining sections describe the required preventative maintenance procedures.

INSTRUCTIONS

2.0 Floppy Disk Drive Preventive Maintenance

(NOTE TO PERFORM ANY OF THE FOLLOWING THE POWER MUST BE REMOVED. The equipment described in this bulletin contains hazardous voltages capable of inflicting personal injury. The cautionary and warning notes are included in this bulletin to alert operator and maintenance personnel to these possible hazards.)

2.1 Preventive Maintanance Equipment

2.1.1 A Head Cleaning Kit consisting of Diskette made of lint-free gauze and denatured alcohol. (Recommended kit: Inmac Clean Cycle

Master Kit No. 7190AC or equiv$\overset{a}{e}$l$\overset{e}{a}$nt.)

2.1.2 A light synthetic oil. (CAUTION. Do not lubricate the drive except as specifically directed. Excess oil will allow dust and dirt to accumulate.) (Recommended kit: Synthetic oil, REMEX P/N 716004-232 or equivelant.)

2.2 Read/Write Head Cleaning. Nestar does not recommend touching or cleaning the Read/Write Head with anything other than a Diskette Head Cleaning Kit. DO NOT AT ANY TIME ATTEMPT TO USE A Q-TIP AND ALCOHOL. Use of the foremention objects will damage the Read/Write Head. You may contact your dealer for information concerning a recommended Head Cleaning Kit. The following procedure is recommended:

2.2.1 Upon obtaining a Head Cleaning Kit the following steps should be adhered to: Enter the NESTAR File Server Command mode and execute the FORMAT program. At this point insert the Head Cleaning Diskette into Drive #1. Follow the procedure to physically format the diskette. You will answer NO to : ENTER Y TO VERIFY ALL DATA ON THE DISK (Y/N) and ENTER Y TO INITIALIZE THE DISK ROOT DIRECTORY (Y/N). When asked "ENTER Y TO PHYSICALLY FORMAT ANOTHER DISK (Y/N): enter Y. Insert the diskette into the Drive #2 and repeat the above procedure only this time for Drive #2. Upon completion of this procedure, you may remove the diskette and return the system to its normal mode of operation.

(CAUTION. Do not smoke while cleaning. Do not touch the face of the heads with your fingers. It is important when performing maintenance on the drive that the heads are not touched or disturbed. Do not attempt to push the head load bail to see how the head works; this causes the top and bottom heads to bang into one another. (See Diagram 2.0.) Do not leave residue or lint on the head face. Trapped residual particles can result in the loss of a head/or a scored diskette.)

2.3 Drive Belt. (This procedure should be attempted by maintenance personnel only.) Check for frayed or weakened areas. To obtain access to the drive belt, you will have to remove the top cover so you may also remove the three (3) connections at the rear of the drive. To remove the cover you will have to remove the two (2) screws to the rear and upper portion of the unit. The cover may then be removed by raising the rear portion and them sliding the cover towards the rear of the machine. In removing the connections they are designed so you may only put them back on correctly. At this point you must remove the four (4) screws under the the base of the drive. This will allow you then to slide the drive out of the unit. Upon removal be extremely careful not knock off any of the connectors on the bottom of the drive. (See Diagram 2.1.)

2.4 Base. Inspect for cleanliness. Clean dust from base and check for loose screws. To inspect the base you will have to remove the top cover. Removal of the two (2) screws to the rear and upper portion of the unit will allow you to remove the cover. Upon removal of the cover, inspect the inside of the unit for dust build-up and loose screws. If found to be dirty, clean with a small vacuum or air hose. Check unit for loose screws, if any are found, replace in appropiate spots.